

CLAIMS

What is claimed is:

- 1 1. A method comprising:
 - 2 parameterizing a routing policy; and
 - 3 applying the parameterized-routing policy to a route.
- 1 2. The method of claim 1 wherein the routing policy comprises a plurality of policy statements, and wherein parameterizing comprises assigning parameters to at least some of the policy statements and refraining from assigning parameters to at least some other of the policy statements to generate the parameterized-routing policy.
- 1 3. The method of claim 1 wherein parameterizing comprises:
 - 2 for the routing policy, generating at least one parameterized-policy statement having an associated set of parameters for one of either a customer or customer class.
- 1 4. The method of claim 1 wherein the routing policy comprises a plurality of policy statements, each policy statement having one or more differing values associated with one or more customers or customer classes, and
4 wherein parameterizing comprises assigning parameters to the one or more differing values of the policy statements.
- 1 5. The method of claim 1 wherein parameterizing further comprises:
 - 2 identifying one or more common blocks of policy statements within the policy;
 - 4 assigning sets of parameters to elements of the one or more common blocks; and
 - 6 storing the parameter sets in a parameter table, the table associating each set of parameters with either the customer or the customer class.

1 6. The method of claim 5 wherein parameterizing further comprises
2 reusing the common blocks in the parameterized-routing policy.

1 7. The method of claim 6 wherein parameterizing further comprises
2 reusing the common blocks in another parameterized-routing policy.

1 8. The method of claim 6 wherein reusing the common blocks comprises
2 calling a parameterized policy with parameters from the parameter table based on
3 one of either the customer or the customer class.

1 9. The method of claim 5 wherein applying further comprises determining
2 at least one of whether to accept the route, whether to modify attributes of the
3 route, or whether to send the route or the modified route to peer routing systems.

1 10. The method of claim 9 wherein when the route is accepted or
2 modified, applying further comprises installing the accepted or the modified route.

1 11. The method of claim 9 further comprising modifying attributes of the
2 route, wherein modifying comprises at least one of changing an attribute, creating
3 a new attribute, or deleting an attribute of the route.

1 12. The method of claim 1 further comprising:
2 identifying one or more common blocks of policy statements, the common
3 blocks being common to more than one routing policy;

4 generating a commonized routing policy from the one or more common
5 blocks; and

6 reusing the commonized routing policy by calling the commonized routing
7 policy from within the more than one routing policy which uses the common
8 blocks.

1 13. The method of claim 12 wherein parameterizing comprises assigning
2 parameters to at least some of the policy statements of the common blocks to
3 parameterize at least some policy statements in the common blocks.

1 14. A routing apparatus comprising:
2 a processor to parameterize a routing policy and apply the parameterized-
3 routing policy to a received route; and
4 a storage element to store parameters associated with the parameterized-
5 routing policy.

1 15. The apparatus of claim 14 wherein the routing policy comprises a
2 plurality of policy statements, and wherein the processor is to assign parameters to
3 at least some of the policy statements and is to refrain from assigning parameters
4 to at least some other of the policy statements to generate the parameterized-
5 routing policy,
6 and wherein the processor is to store the assigned parameters in the storage
7 element.

1 16. The apparatus of claim 14 wherein the processor is to generate at least
2 one parameterized-policy statement having an associated set of parameters for one
3 of either a customer or customer class.

1 17. The apparatus of claim 14 wherein the routing policy comprises a
2 plurality of policy statements, each policy statement having one or more differing
3 values associated with one or more customers or customer classes, and
4 wherein the processor is to assign parameters to the one or more differing
5 values of the policy statements.

1 18. The apparatus of claim 14 wherein the processor is to
2 identify one or more common blocks of policy statements within the
3 policy;
4 assign sets of parameters to elements of the one or more common blocks;
5 and
6 store the parameter sets in a parameter table of the storage element, the
7 table associating each set of parameters with either the customer or the customer
8 class.

1 19. The apparatus of claim 18 wherein the processor is to reuse the
2 common blocks in the parameterized-routing policy.

1 20. The apparatus of claim 19 wherein the processor is to reuse the
2 common blocks in another parameterized-routing policy.

1 21. The apparatus of claim 19 wherein the processor, as part of reusing, is
2 to call a parameterized policy with parameters from the parameter table based on
3 one of either the customer or the customer class.

1 22. The apparatus of claim 18 wherein the processor is to determine at
2 least one of whether to accept the route, whether to modify attributes of the route,
3 or whether to send the route or the modified route to peer routing systems.

1 23. The apparatus of claim 22 wherein when the route is accepted or
2 modified, the processor is to install the accepted or the modified route on a router.

1 24. The apparatus of claim 22 wherein the processor is to modify attributes
2 of the route by at least one of changing an attribute, creating a new attribute, or
3 deleting an attribute of the route.

1 25. The apparatus of claim 14 wherein the processor is to further:
2 identify one or more common blocks of policy statements, the common
3 blocks being common to more than one routing policy;
4 generate a commonized routing policy from the one or more common
5 blocks; and
6 reuse the commonized routing policy by calling the commonized routing
7 policy from within the more than one routing policy which uses the common
8 blocks.

1 26. The apparatus of claim 25 wherein the processor is to assign
2 parameters to at least some of the policy statements of the common blocks to
3 parameterize the at least some policy statements in the common blocks.

1 27. A system comprising:
2 means for parameterizing a routing policy;
3 means for applying the parameterized-routing policy to a received route;
4 and
5 means for storing parameters associated with the parameterized-routing
6 policy.

1 28. The system of claim 27 wherein the routing policy comprises a
2 plurality of policy statements, and wherein the means for parameterizing is to
3 assign parameters to at least some of the policy statements and is to refrain from
4 assigning parameters to at least some other of the policy statements to generate the
5 parameterized-routing policy,
6 and wherein the means for parameterizing is to store the assigned
7 parameters in the means for storing.

1 29. The system of claim 27 wherein the means for parameterizing is to
2 generate at least one parameterized-policy statement having an associated set of
3 parameters for one of either a customer or customer class.

1 30. The system of claim 27 wherein the routing policy comprises a
2 plurality of policy statements, each policy statement having one or more differing
3 values associated with one or more customers or customer classes, and
4 wherein the means for parameterizing is to assign parameters to the one or
5 more differing values of the policy statements.

1 31. The system of claim 27 wherein the means for parameterizing is to
2 further:
3 identify one or more common blocks of policy statements within the
4 policy;

5 assign sets of parameters to elements of the one or more common blocks;

6 and

7 store the parameter sets in a parameter table of the storage element, the
8 table associating each set of parameters with either the customer or the customer
9 class.

1 32. The system of claim 31 wherein the means for applying is to reuse the
2 common blocks in the parameterized-routing policy.

1 33. The system of claim 32 wherein the means for applying is to reuse the
2 common blocks in another parameterized-routing policy.

1 34. The system of claim 32 wherein the means for applying, as part of
2 reusing, is to call a parameterized policy with parameters from the parameter table
3 based on one of either the customer or the customer class.

1 35. The system of claim 31 wherein the means for applying is to determine
2 at least one of whether to accept the route, whether to modify attributes of the
3 route, or whether to send the route or the modified route to peer routing systems.

1 36. The system of claim 35 wherein when the route is accepted or
2 modified, the means for applying is to install the accepted or the modified route on
3 a router.

1 37. The system of claim 35 wherein the means for applying is to modify
2 attributes of the route by at least one of changing an attribute, creating a new
3 attribute, or deleting an attribute of the route.

1 38. The system of claim 27 wherein the means for parameterizing is to:
2 identify one or more common blocks of policy statements, the common
3 blocks being common to more than one routing policy; and
4 generate a commonized routing policy from the one or more common
5 blocks, and

6 wherein the means for applying is to reuse the commonized routing policy
7 by calling the commonized routing policy from within the more than one routing
8 policy which uses the common blocks.

1 39. The system of claim 38 wherein the means for parameterizing is to
2 assign parameters to at least some of the policy statements of the common blocks
3 to parameterize the at least some policy statements in the common blocks.

1 40. A machine-readable medium that provides instructions, which when
2 executed by one or more processors, cause said processors to perform operations
3 comprising:

4 parameterizing a routing policy; and
5 applying the parameterized-routing policy to a route.

1 41. The machine-readable medium of claim 40 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations, wherein the routing policy comprises a plurality of policy
4 statements, and wherein parameterizing comprises assigning parameters to at least
5 some of the policy statements and refraining from assigning parameters to at least
6 some other of the policy statements to generate the parameterized-routing policy.

1 42. The machine-readable medium of claim 40 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations, wherein parameterizing comprises: for the routing policy,
4 generating at least one parameterized-policy statement having an associated set of
5 parameters for one of either a customer or customer class.

1 43. The machine-readable medium of claim 40 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations wherein the routing policy comprises a plurality of policy
4 statements, each policy statement having one or more differing values associated
5 with one or more customers or customer classes.

1 44. The machine-readable medium of claim 43 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations wherein parameterizing comprises assigning parameters to the
4 one or more differing values of the policy statements.

1 45. The machine-readable medium of claim 40 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations comprising:

4 identifying one or more common blocks of policy statements within the
5 policy;

6 assigning sets of parameters to elements of the one or more common
7 blocks; and

8 storing the parameter sets in a parameter table, the table associating each
9 set of parameters with either the customer or the customer class.

1 46. The machine-readable medium of claim 45 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations comprising reusing the common blocks in the parameterized-
4 routing policy.

1 47. The machine-readable medium of claim 45 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations comprising reusing the common blocks in another
4 parameterized-routing policy.

1 48. The machine-readable medium of claim 46 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations comprising reusing the common blocks comprises calling a
4 parameterized policy with parameters from the parameter table based on one of
5 either the customer or the customer class.

1 49. The machine-readable medium of claim 45 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to

3 perform operations comprising applying further comprises determining at least
4 one of whether to accept the route, whether to modify attributes of the route, or
5 whether to send the route or the modified route to peer routing systems.

1 50. The machine-readable medium of claim 40 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations further comprising:

4 identifying one or more common blocks of policy statements, the common
5 blocks being common to more than one routing policy;
6 generating a commonized routing policy from the one or more common
7 blocks; and

8 reusing the commonized routing policy by calling the commonized routing
9 policy from within the more than one routing policy which uses the common
10 blocks.

1 51. A method for implementing routing policy comprising:
2 performing a policy translation on a policy configuration to generate an
3 internal-policy representation;
4 verifying attribute-operator pairings of the internal-policy representation
5 with one or more client dynamic load libraries (DLLs); and
6 when the attribute-operator pairings have been verified, compiling the
7 internal-policy representation and storing the complied internal-policy
8 representation in a system database.

1 52. The method as claimed in claim 51 further comprising notifying a
2 client protocol that the routing policy has been modified.

1 53. The method as claimed in claim 52 further comprising, after notifying,
2 applying the new policy to a route received from the client protocol.

1 54. The method as claimed in claim 51 further comprising receiving the
2 policy configuration in the form of one or more policy statements.

1 55. The method as claimed in claim 51 further comprising receiving the
2 policy configuration entered by a user or system administrator in the form of one
3 or more policy statements, the policy statements being in grammatical form.

1 56. The method as claimed in claim 51 wherein verifying comprises
2 verifying each of a plurality of attribute-operator pairings of the internal-policy
3 representation with the client DLLs.

1 57. The method as claimed in claim 56 wherein the client DLLs include
2 verification routines for the attributes and associated operators to allow a policy
3 repository to query each attribute-operator pairing and to verify arguments present
4 in the configuration for each statement in the policy.

1 58. The method as claimed in claim 51 wherein compiling further
2 comprises generating the compiled internal-policy representation in a policy
3 transmission language (PXL).

1 59. The method as claimed in claim 51 further comprising parameterizing
2 the routing policy.

1 60. A system for generating routing policy comprising:
2 a translator to perform a policy translation on a policy configuration to
3 generate an internal-policy representation; and
4 a policy repository to verify attribute-operator pairings of the internal-
5 policy representation with one or more client dynamic load libraries (DLLs), the
6 policy repository to compile the internal-policy representation when the attribute-
7 operator pairings are verified.

1 61. The system of claim 60 wherein the policy repository is to verify
2 arguments of the attribute-operator pairings.

1 62. The system as claimed in claim 60 further comprising a router to apply
2 the new policy to a route received from the client protocol.

1 63. The system as claimed in claim 60 further comprising an I/O device to
2 receive the policy configuration in the form of one or more policy statements.

1 64. The system as claimed in claim 60 further comprising an I/O device to
2 receive the policy configuration entered by a system administrator or user in the
3 form of one or more policy statements, the policy statements being in a
4 grammatical form.

1 65. The system as claimed in claim 60 wherein the policy repository is to
2 verify each of a plurality of attribute-operator pairings of the internal-policy
3 representation with the client DLLs, and is to further verify arguments of the
4 pairings with the client DLLs.

1 66. The system of claim 65 wherein the client DLLs comprise verification
2 routines for the attributes and associated operators to allow the policy repository to
3 query each attribute-operator pairing and to verify arguments present in the
4 configuration with each statement in the policy.

1 67. The system as claimed in claim 60 wherein the policy repository, as
2 part of compiling, is to generate the compiled internal-policy representation in a
3 policy transmission language (PXL).

1 68. The system as claimed in claim 60 further comprising a system
2 database to store the complied internal-policy representation.

1 69. A machine-readable medium that provides instructions, which when
2 executed by one or more processors, cause said processors to perform operations
3 comprising:

4 performing a policy translation on a policy configuration to generate an
5 internal-policy representation;

6 verifying attribute-operator pairings of the internal-policy representation
7 with one or more client dynamic load libraries (DLLs); and

8 when the attribute-operator pairings are verified, compiling the internal-
9 policy representation and storing the complied internal-policy representation in a
10 system database.

1 70. The machine-readable medium of claim 69 wherein the instructions,
2 when further executed by the said processors result in:
3 receiving the policy configuration entered by a user or system
4 administrator in the form of one or more policy statements, the policy statements
5 being in grammatical form;
6 notifying a client protocol that the routing policy has been modified; and
7 applying the new policy to a route received from the client protocol

1 71. The machine-readable medium of claim 70 wherein the instructions,
2 when further executed by said processors result in:
3 verifying each of a plurality of attribute-operator pairings of the internal-
4 policy representation with the client DLLs, wherein the client DLLs include
5 verification routines for the attributes and associated operators to allow a policy
6 repository to query each attribute-operator pairing and to verify arguments present
7 in the configuration to each statement in the policy; and
8 generating the compiled internal-policy representation in a policy
9 transmission language (PXL).

1 72. A system for generating routing policy comprising:
2 means for performing a policy translation on a policy configuration to
3 generate an internal-policy representation;
4 means for verifying attribute-operator pairings of the internal-policy
5 representation with one or more client dynamic load libraries (DLLs);
6 means for compiling the internal-policy representation when the attribute-
7 operator pairings verify; and
8 means for notifying a client protocol that the routing policy has been
9 modified.

1 73. The system of claim 72 further comprising means for verifying
2 arguments of the attribute-operator pairings;

1 74. The system of claim 72 further comprising means for applying the new
2 policy to a route received from the client protocol.

1 75. The system of claim 72 further comprising means for altering
2 application state.

1 76. The system of claim 72 further comprising means for logging events.

1 77. The system of claim 72 further comprising means for altering
2 configuration values.

1 78. A method of altering route attributes in a running network comprising:
2 measuring characteristics available to the system, the characteristics
3 including route and non-route related characteristics; and
4 modifying route attributes of an existing route based on the measured
5 characteristic.

1 79. The method of claim 78 further comprising tagging the existing route
2 when a characteristic exceeds a predetermined measurement amount, and
3 wherein modifying comprises dynamically modifying the route attributes
4 based on the tagging.

1 80. The method of claim 79, wherein the characteristic includes a traffic
2 statistic including at least one of either bandwidth or a number of dropped packets
3 of a link.

1 81. A routing system comprising:
2 a traffic evaluator to measure characteristics available to the system, the
3 characteristics including route and non-route related characteristics; and

4 a processing element to modify route attributes of an existing route based
5 on the measured characteristic.

1 82. The routing system of claim 81 wherein the traffic evaluator is to tag
2 the existing route when a characteristic exceeds a predetermined measurement
3 amount, and
4 wherein the processing element is to dynamically modify the route
5 attributes based on the tagging.

1 83. The routing system of claim 82, wherein the characteristic includes a
2 traffic statistic including at least one of either bandwidth or a number of dropped
3 packets of a link.

1 84. A routing system comprising:
2 means for measuring characteristics available to the system, the
3 characteristics including route and non-route related characteristics; and
4 means for modifying route attributes of an existing route based on the
5 measured characteristic.

1 85. The system of claim 84 further comprising means for tagging the
2 existing route when a characteristic exceeds a predetermined measurement
3 amount, and
4 wherein the means for modifying comprises means for dynamically
5 modifying the route attributes based on the tagging.

1 86. The method of claim 84, wherein the characteristic includes a traffic
2 statistic including at least one of either bandwidth or a number of dropped packets
3 of a link.

1 87. A machine-readable medium that provides instructions, which when
2 executed by one or more processors, cause said processors to perform operations
3 comprising:

4 measuring characteristics available to the system, the characteristics
5 including route and non-route related characteristics; and
6 modifying route attributes of an existing route based on the measured
7 characteristic.

1 88. The machine-readable medium of claim 87 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations further comprising tagging the existing route when a
4 characteristic exceeds a predetermined measurement amount, and
5 wherein modifying comprises dynamically modifying the route attributes
6 based on the tagging.

1 89. The machine-readable medium of claim 88 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations, wherein the characteristic includes a traffic statistic including
4 at least one of either bandwidth or a number of dropped packets of a link.

1 90. A method of configuring a router comprising:
2 evaluating a configuration file to determine relationships between route
3 attributes;
4 expressing the relationships in one or more mathematical functions; and
5 when a route is received from a customer, modifying the attributes of the
6 received route based on results of the mathematical functions.

1 91. The method of claim 90 further comprising:
2 generating a revised configuration file which expresses the relationships
3 between the route attributes with the one or more mathematical functions; and
4 wherein when the route is received from the customer, modifying
5 comprises modifying the attributes of the received route using the revised
6 configuration file.

1 92. The method of claim 90 further comprising routing a received packet
2 using the received route with the modified attributes.

1 93. The method of claim 90 wherein the one or more mathematical
2 functions comprises at least one of:
3 adding an offset to a value of a first attribute for a determined value of a
4 second attribute; and
5 multiplying the value of the first attribute by a factor for the determined
6 value of the second attribute.

1 94. The method of claim 93 wherein the first and second attributes
2 comprise at least two of a community tag attribute, a local preference attribute,
3 and a multi-exit discriminator (MED) attribute, and
4 wherein evaluating determines relationships between the first attribute and
5 the second attribute of routes in the configuration file, and expressing comprises
6 generating a mathematical relation between the first and second attributes.

1 95. A routing system comprising:
2 a processor to evaluate a configuration file to determine relationships
3 between route attributes and to generate the relationships in one or more
4 mathematical functions; and
5 a storage element to store relationships express in the one or more
6 mathematical functions, wherein when a route is received from a customer, the
7 processor is to modify the attributes of the received route based on results of the
8 mathematical functions.

1 96. The system of claim 95 wherein the storage element is to store a
2 revised configuration file which expresses the relationships between the route
3 attributes with the one or more mathematical functions.

1 97. The system of claim 95 wherein the processor is to route a received
2 packet using the received route with the modified attributes.

1 98. The system of claim 95 wherein the one or more mathematical
2 functions comprises at least one of:

3 adding an offset to a value of a first attribute for a determined value of a
4 second attribute; and

5 multiplying the value of the first attribute by a factor for the determined
6 value of the second attribute.

1 99. The system of claim 98 wherein the first and second attributes
2 comprise at least two of a community tag attribute, a local preference attribute,
3 and a multi-exit discriminator (MED) attribute, and

4 wherein the processor is to determine relationships between the first
5 attribute and the second attribute of routes in the configuration file, and is to
6 generate a mathematical relation between the first and second attributes.

1 100. A system comprising:

2 means for evaluating a configuration file to determine relationships
3 between route attributes;

4 means for expressing the relationships in one or more mathematical
5 functions; and

6 means for modifying the attributes of a received route based on the
7 mathematical functions.

1 101. The system of claim 100 further comprising:

2 means for generating a revised configuration file which expresses the
3 relationships between the route attributes with the one or more mathematical
4 functions.

1 102. The system of claim 100 further comprising means for routing a
2 received packet using the received route with the modified attributes.

1 103. The system of claim 100 wherein the one or more mathematical
2 functions comprises at least one of:

3 an addition function for adding an offset to a value of a first attribute for a
4 determined value of a second attribute; and

5 a multiplication function for multiplying the value of the first attribute by a
6 factor for the determined value of the second attribute.

1 104. The system of claim 103 wherein the first and second attributes
2 comprise at least two of a community tag attribute, a local preference attribute,
3 and a multi-exit discriminator (MED) attribute, and
4 wherein the means for evaluating is to determine relationships between the
5 first attribute and the second attribute of routes in the configuration file, and is to
6 generate a mathematical relation between the first and second attributes.

1 105. A machine-readable medium that provides instructions, which when
2 executed by one or more processors, cause said processors to perform operations
3 comprising:

4 evaluating a configuration file to determine relationships between route
5 attributes;
6 expressing the relationships in one or more mathematical functions; and
7 when a route is received from a customer, modifying the attributes of the
8 received route based on the mathematical functions.

1 106. The machine-readable medium of claim 105 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations further comprising:

4 generating a revised configuration file which expresses the relationships
5 between the route attributes with the one or more mathematical functions; and
6 wherein when the route is received from the customer, modifying
7 comprises modifying the attributes of the received route using the revised
8 configuration file.

1 107. The machine-readable medium of claim 105 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations further comprising routing a received packet using the
4 received route with the modified attributes, and wherein the one or more
5 mathematical functions comprises at least one of:

6 adding an offset to a value of a first attribute for a determined value of a
7 second attribute; and

8 multiplying the value of the first attribute by a factor for the determined
9 value of the second attribute.

1 108. The machine-readable medium of claim 107 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations, wherein the first and second attributes comprise at least two
4 of a community tag attribute, a local preference attribute, and a multi-exit
5 discriminator (MED) attribute, and

6 wherein evaluating determines relationships between the first attribute and
7 the second attribute of routes in the configuration file, and expressing comprises
8 generating a mathematical relation between the first and second attributes.

1 109. A method comprising:

2 applying at least one of a plurality of event triggers to a received route; and
3 taking a predetermined action when the applying indicates that a triggering
4 event occurs.

1 110. The method of claim 109 wherein the predetermined action includes
2 one of limiting a number of routes received from a peer, filtering specific blocks
3 of address space on the peer may be filtered, or logging a message.

1 111. The method of claim 110 further comprising applying policy to the
2 received route, and wherein the event triggers comprise an extension to language
3 of the policy.

1 112. A routing system comprising:

2 a storage element to store a plurality of event triggers; and
3 a processor to apply at least one of the event triggers to a received route,
4 and to take a predetermined action when the applying indicates that a triggering
5 event occurs.

1 113. The system of claim 112 wherein the predetermined action includes
2 one of limiting a number of routes received from a peer, filtering specific blocks
3 of address space on the peer may be filtered, or logging a message.

1 114. The system of claim 113 wherein the processor is to further apply
2 policy to the received route, and wherein the event triggers comprise an extension
3 to language of the policy.

1 115. A system comprising:
2 means applying at least one of a plurality of event triggers to a received
3 route; and
4 means for taking a predetermined action when the means for applying
5 indicates that a triggering event occurs.

1 116. The system of claim 115 wherein the predetermined action includes
2 one of limiting a number of routes received from a peer, filtering specific blocks
3 of address space on the peer may be filtered, or logging a message.

1 117. The system of claim 116 further comprising means for applying
2 policy to the received route, and wherein the event triggers comprise an extension
3 to language of the policy.

1 118. A machine-readable medium that provides instructions, which when
2 executed by one or more processors, cause said processors to perform operations
3 comprising:
4 applying at least one of a plurality of event triggers to a received route; and
5 taking a predetermined action when the applying indicates that a triggering
6 event occurs.

1 119. The machine-readable medium of claim 118 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations, wherein the predetermined action includes one of limiting a

4 number of routes received from a peer, filtering specific blocks of address space
5 on the peer may be filtered, or logging a message.

1 120. The machine-readable medium of claim 119 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations further comprising applying policy to the received route, and
4 wherein the event triggers comprise an extension to language of the policy.

1 121. A method of staging routing policies comprising:
2 applying a staged-routing policy and an existing-routing policy to a
3 received route;
4 updating a routing-information base with results of the applied staged-
5 routing policy and the applied existing-routing policy; and
6 comparing packet forwarding results of the staged and existing routing
7 policies by applying the updated routing-information base to packets.

1 122. The method of claim 121 wherein updating comprises marking
2 updates from the applied-staged routing policy in the routing-information base.

1 123. A routing system comprising:
2 a storage element to store a routing-information base, a staged-routing
3 policy and an existing-routing policy; and
4 a processor to apply a staged-routing policy and an existing-routing policy
5 to a received route, to update the routing-information base with results of the
6 applied staged-routing policy and the applied existing-routing policy, and to
7 compare packet forwarding results of the staged and existing routing policies by
8 applying the updated routing-information base to packets.

1 124. The system of claim 123 wherein the processor is mark updates from
2 the applied-staged routing policy in the routing-information base.

1 125. A system comprising:

2 means for applying a staged-routing policy and an existing-routing policy
3 to a received route;

4 means for updating a routing-information base with results of the applied
5 staged-routing policy and the applied existing-routing policy; and

6 means for comparing packet forwarding results of the staged and existing
7 routing policies by applying the updated routing-information base to packets.

1 126. The system of claim 125 wherein the means for updating further
2 comprises means for marking updates from the applied-staged routing policy in
3 the routing-information base.

1 127. A machine-readable medium that provides instructions, which when
2 executed by one or more processors, cause said processors to perform operations
3 comprising:

4 applying a staged-routing policy and an existing-routing policy to a
5 received route;

6 updating a routing-information base with results of the applied staged-
7 routing policy and the applied existing-routing policy; and

8 comparing packet forwarding results of the staged and existing routing
9 policies by applying the updated routing-information base to packets.

1 128. The machine-readable medium of claim 127 wherein the instructions,
2 when further executed by one or more of said processors cause said processors to
3 perform operations, wherein updating comprises marking updates from the
4 applied-staged routing policy in the routing-information base.

1 129. A method of generating routing policy comprising:

2 decoupling data items from actions associated with the data items; and

3 building a routing policy by combining common components using set
4 operations on the data items and the actions,

5 wherein common data items that span across more than one domain are
6 reused for the more than one domain,

7 wherein the data items comprise prefixes, as-path expressions and
8 community lists, and
9 wherein the actions comprise at least accept and deny actions.

1 130. A routing system comprising:
2 a plurality of filters that share common data items for reuse across more
3 than one domain; and
4 one or more processing elements to decouple data items from actions
5 associated with the data items and to build a routing policy by combining common
6 components using set operations on the data items and the actions,
7 wherein common data items that span across more than one domain are
8 reused for the more than one domain,
9 wherein the data items comprise prefixes, as-path expressions and
10 community lists,
11 wherein the actions comprise at least accept and deny actions, and
12 wherein the plurality of filters comprise either packet or route filters.

1 131. A method of generating routing policy comprising:
2 prefixing statements of a single routing policy with either source or sink
3 designations;
4 applying statements with the source designations to match operations; and
5 applying statements with the sink designations to set operations.

1 132. A routing system comprising:
2 a single routing policy comprising statements prefixed with either source
3 or sink designations; and
4 processing elements to apply ones of the statements with the source
5 designations to match operations, and to apply one of the statements with the sink
6 designations to set operations.